Topology Design of Network-Coding-Based Multicast Networks

Kaikai Chi, *Student Member*, *IEEE*, Xiaohong Jiang, *Member*, *IEEE*, Susumu Horiguchi, *Senior Member*, *IEEE*, and Minyi Guo, *Senior Member*, *IEEE*

Abstract—It is anticipated that a large amount of multicast traffic needs to be supported in future communication networks. The network coding technique proposed recently is promising for establishing multicast connections with a significantly lower bandwidth requirement than that of traditional Steiner-tree-based multicast connections. How to design multicast network topologies with the consideration of efficiently supporting multicast by the network coding technique becomes an important issue now. It is notable, however, that the conventional algorithms for network topology design are mainly unicast-oriented, and they cannot be adopted directly for the efficient topology design of network-coding-based multicast networks by simply treating each multicast as multiple unicasts. In this paper, we consider for the first time the novel topology design problem of network-coding-based multicast networks. Based on the characteristics of multicast and network coding, we first formulate this problem as a mixed-integer nonlinear programming problem, which is NP-hard, and then propose two heuristic algorithms for it. The effectiveness of our heuristics is verified through simulation and comparison with the exhaustive search method. We demonstrate in this paper that, in the topology design of multicast networks, adopting the network coding technique to support multicast transmissions can significantly reduce the overall topology cost as compared to conventional unicast-oriented design and the Steiner-tree-based design.

Index Terms—Network coding, multicast networks, topology design, heuristic algorithms.

1 Introduction

With the advance of communication networks, a great number of multicast applications such as video conferencing have emerged, and it is foreseeable that more multicast applications will emerge in the near future. As many multicast services require the transmission of video streaming traffic, future networks will need to support a considerable amount of multicast traffic.

Network coding is a novel technique proposed to save bandwidth and increase the throughput of multicast communication [1]. In existing networks, each node either forwards packets directly (in unicast transmissions) or replicates packets and forwards them (in multicast transmissions). Network coding generalizes the traditional routing approach by allowing network nodes to generate new packets by performing algebraic operations on packets received over the incoming links. The principle of network coding can be easily explained by considering a simple multicast example (from [1]) shown in Fig. 1. All links are error-free and have a capacity of 1 bit per unit time. Source node s has to transmit data to sink nodes t_1 and t_2 at the rate of 2 bits per unit time. We can see that this network problem can be satisfied if node c can perform network coding, as shown in Fig. 1, but cannot be satisfied by only forwarding bits at intermediate nodes.

Network coding was originally proposed for a single multicast connection, but it was shown later that network coding can offer advantages for other connection cases as well [2], [3], [4]. Establishing an efficient multicast connection is one of the central problems in network coding. Via the network coding technique, multicast connections can be established with significantly lower bandwidth consumption than that consumed by Steiner-tree-based multicast transmissions [1], [5]. In addition to bandwidth saving, network coding can also bring other benefits to multicast connections, such as a significant increase in the throughput of multicast connections [6], [7] and an improvement of system robustness and adaptability [8], [9].

Owing to its high capability to efficiently support multicast transmissions, the network coding technique is promising to be applied in future multicast networks. Consequently, network-coding-based multicast (NCM) network design with the consideration of efficiently supporting multicast by the network coding technique becomes an important issue now. A complete network design involves a lot of aspects such as traffic matrix estimation, topology design, node function specification, and management [10]. Topology design is one of the most important aspects of network design.

Network topology design has long been a challenging problem. Given the number of nodes, physical locations of these nodes, knowledge of communication lines available, and traffic requirements, topology design is to assign communication links, the capacity of each link, and the flow of each traffic requirement. These assignments should keep the resulting topology cost as low as possible while satisfying a set of requirements such as the delay requirement and reliability requirement. The topology

- *K. Chi, X. Jiang, and S. Horiguchi are with the Horiguchi Lab, Graduate School of Information Sciences, Tohoku University, Aobayama 6-3-09, Sendai, 980-8579, Japan. E-mail: {kai-chi, jiang, susumu}@ecei.tohoku.ac.jp.*
- *M. Guo is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: minyi@u-aizu.ac.jp.*

Manuscript received 28 Sept. 2006; revised 19 Feb. 2007; accepted 21 June 2007; published online 11 July 2007.
Recommended for acceptance by S. Das.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0304-0906.
Digital Object Identifier no. 10.1109/TPDS.0304.0906.

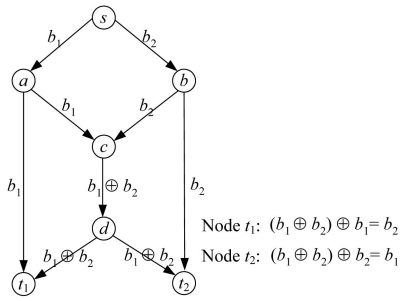

Fig. 1. A one-source two-sink network with coding.

optimization problem is generally an NP-hard combinatorial optimization problem [11], [12] and quickly becomes intractable as the number of nodes increases. Conventional topology design problems only considered unicast requirements due to the fact that, at that time, there were few or no multicast applications. So far, a number of unicast-oriented heuristic algorithms have been proposed to deal with specific topology design problems, including some classic ones such as Branch Exchange, Cut Saturation, and the MENTOR Algorithm [12], [13], [14] and some modern ones such as Tabu Search, Simulated Annealing, and the Genetic Algorithm [15], [16], [17].

The topology design problem of NCM networks is based on the assumption that network nodes have the capability of performing encoding and are more difficult than traditional ones. Two aspects distinguish this problem from conventional ones. First, multicast requirements are considered in this problem. Second, the network coding technique is applied to support multicast transmissions. The consideration of network-coding-based multicast increases the complexity of the optimal routing subproblem and the corresponding topology design problem because the NCM routing complexity is much higher than that of the unicast case [5] and the routing procedure must be embedded in topology design algorithms. Therefore, effective topology design heuristics should be developed for NCM networks. How to take full advantage of the characteristics of network-coding-based multicast to save bandwidth in the design process and at the same time keep the algorithm complexity as low as possible is the challenge topology designers have to face.

In this paper, we consider the topology design problem of NCM networks. The main contributions of this paper are summarized as follows:

1. For the first time, we formally formulate the optimal topology design of NCM networks as a mixed-integer programming problem, which is NP-hard. The mathematical formulation can help us assess the essence and understand the hardness of this problem well.
2. Two heuristic algorithms, the *link deletion and exchange (LDE) algorithm* and the *link addition and exchange (LAE) algorithm*, are proposed for the efficient topology design of NCM networks.
3. We demonstrate that, by adopting the network coding technique to support multicast transmissions, we can design a multicast network topology with a significantly lower network cost than that of the conventional unicast-oriented and Steiner-tree-based designs.

The paper is organized as follows: Section 2 defines the topology design problem and formally formulates it as a mixed-integer nonlinear programming problem. Two heuristic algorithms for our NCM topology design problem are introduced in Section 3. Section 4 presents simulation results to evaluate their performance and demonstrates the benefit offered by the network coding technique in network topology design as well. Finally, Section 5 concludes the paper.

2 Problem Statement and Formulation

Some important notions in the network topology design are listed as follows:

- **Traffic requirement.** This refers to the average number of bits per second sent from a source to a destination or a set of destinations.
- **Network reliability.** This refers to the reliability of the overall network to provide communication in the event of failure of a component or components in the network.
- **Topological configuration** (for simplicity, called configuration). This refers to the set of links connecting network nodes together.
- **Capacity assignment.** This refers to the determination of the maximum number of bits per second that can be transmitted by each communication link of a given configuration.
- **Flow assignment.** This refers to the selection of the route for each traffic requirement.
- **The average packet delay.** This refers to the mean time taken by a packet to travel from a source node to a destination node.

In the available literature, almost all network topology design problems are unicast oriented. Two significant aspects distinguish the topology design problem of NCM networks from old ones. First, multicast requirements are considered specially. Second, the network coding technique is applied to support multicast transmissions. The specific topology design problem of NCM networks we consider can be stated as follows:

Given

1. the number of nodes N and their corresponding locations,
2. the unicast traffic requirement between each ordered pair of distinct nodes,
3. the source node, destination nodes, and multicast rate of each multicast traffic requirement,
4. the capacity, fixed cost, and cost per unit length of each type of communication line (that is, cable),
5. the reliability requirement, and
6. the delay requirement,

Minimize the overall topology cost

Over

1. all possible configurations,
2. all possible link capacity assignments, and
3. all possible flow assignments,

TABLE 1
Notation Used in This Paper

Notation	Meaning
$\mathcal{G}(\mathcal{N},\mathcal{A})$	Directed graph consisting of a node set $\mathcal{N}$ and an arc set $\mathcal{A}$.
$\{i,j\}$	An unordered node pair called a link. It is the same as $\{j,i\}$.
(i,j)	An ordered node pair called an arc.
N	Number of network nodes.
$d_{i,j}$	Distance between node i and node j. It is the same as $d_{j,i}$.
K	Number of available communication line types.
C_i	Capacity of i-type line ($C_1 < C_2 < \cdots < C_K$).
f_i	Fixed cost of i-type line.
p_i	Cost per unit length of i-type line.
k	Node-connectivity.
e_{max}	Maximum link utilization constraint.
$r_{i,j}$	Traffic requirement rate from node i to node j.
$f_{i,j}^{(i_1,i_2)}$	Amount of unicast flow from node i_1 to i_2 on arc (i,j).
M	Number of multicast traffic requirements.
R_i	Traffic rate of ith multicast requirement.
S_i	Node set of ith multicast requirement. Denote by $n_{i,0}$ the source node, and denote by $n_{i,1},\cdots,n_{i,\lvert S_i\rvert-1}$ destination nodes in this node set.
$g_{i,j}^{(n_{i,0},n_{i,j})}$	Amount of flow from $n_{i,0}$ to $n_{i,j}$ on arc (i,j).
$f_{i,j}$	Total amount of flow on arc (i,j).
$C_{i,j}$	Assigned capacity of link $\{i,j\}$. It is the same as $C_{j,i}$.
$D_{i,j}$	Cost of link $\{i,j\}$. It is the same as $D_{j,i}$.

Subject to

1. capacity assignment constraint,
2. reliability requirement,
3. flow conservation constraint,
4. link utilization (ratio of the used capacity to the total capacity) constraint, and
5. delay requirement.

The notation used in this paper is shown in Table 1. In the remainder of this section, we will deal with different aspects of this problem in detail and finally formulate this problem mathematically.

2.1 Capacity Assignment

Only those types of communication lines that are available in the market can be assigned on network links. Thus, the capacity that can be allocated to a link is the combination of available line capacities. Assume that there are K types of communication lines available, with each type of line having a discrete capacity. Then, the capacities that can be allocated to each network link $\{i,j\}$ are

$$C_{i,j} = u_{i,j}^1 C_1 + \cdots + u_{i,j}^K C_K,$$

where $u_{i,j}^1,\cdots,u_{i,j}^K \in \{0,1,2,\cdots\}$.

2.2 Network Cost Model

The topology cost consists of the material cost of communication lines, installation cost, network node (such as switch) cost, etc. For simplicity, it is often reasonable to approximately model the cost of nodes as fixed line costs and assume that the network cost consists of line costs only. We assume that the cost of placing a line between two nodes comprises two components: a *fixed cost* related to the capacity of this line and a *variable cost* related to the physical length of this line. The fixed cost of a t-type line, f_t, includes the installation cost, the overhead incurred by the endpoints, and so on. The variable cost related to length is linear with line length d and its cost per unit length p_t, that is, it equals $p_t \cdot d$. In addition, the total fixed cost of a network topology usually accounts for a significant percentage of the total cost, and the cost per unit capacity per unit length decreases with the increase of line capacity due to the economy of scale.

For a link $\{i,j\}$, one or more communication lines can be placed on it. Thus, the cost of link $\{i,j\}$ can be expressed as $D_{i,j} = \sum_{t=1}^{K} u_{i,j}^t (f_t + d_{i,j} \cdot p_t)$, where $u_{i,j}^t$ is the number of t-type lines assigned to link $\{i,j\}$. Index N nodes from 1 to N. Then, the overall topology cost is

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} D_{i,j} = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{t=1}^{K} u_{i,j}^t (f_t + d_{i,j} \cdot p_t). \quad (1)$$

2.3 Network Reliability Requirement

Network links and nodes can fail because of different causes. It is necessary to consider the network reliability at the topology design stage. There are different measures to scale the reliability of a network. Here, the concept of k-connectivity is used as the reliability measure. k-connectivity indicates that there are at least k node-disjoint paths available between each pair of nodes. The network is said to be k-node connected if it satisfies the k-connectivity condition.

Define function $F(x)$ as follows: If $x > 0$, $F(x) = 1$; otherwise, $F(x) = 0$.

For each node i, the number of links incident to it is $\sum_{j=1,j\neq i}^{N} F(C_{i,j})$. Then, the k-connectivity requirement for networks can be formulated as follows [18]:

$$\begin{aligned} &\sum_{i\in S}\sum_{j\in\{1,2,\cdots,N\}\setminus(Z\bigcup S)} F(C_{i,j}) \geq 1, \\ &\quad \forall 1 \leq s,d \leq N (s \neq d), \\ &\quad \forall Z \subseteq \mathcal{N}\setminus\{s,d\} \text{ with } |Z| = k-1, \\ &\quad \forall S \subseteq \mathcal{N}\setminus Z \text{ with } s \in S \text{ and } d \notin S. \end{aligned} \quad (2)$$

2.4 Conservation of Flow

The flow conservation law states that, at each node in a communication network, the total incoming flow plus the flow originating at this node minus the demand at this node equals the total outgoing flow. It is easy to understand that unicast flows comply with the flow conservation principle. However, the case of multicast flows is different. At an intermediate node, one ingoing packet of a multicast flow may induce one or several outgoing packets. Thus, multicast flows violate the flow conservation principle. Next, we will consider this issue with unicast, Steiner-tree-based multicast, and network-coding-based multicast, respectively.

2.4.1 Unicast Transmission

For a unicast transmission with rate $r_{s,d}$ from source node s to destination node d, the amount of this unicast traffic into a node must be equal to the amount of this unicast traffic out of this node, unless this node is the source or the destination of this unicast. The flow conservation constraint can be expressed as

$$\sum_{\{j:(i,j)\in\mathcal{A}\}} f_{i,j}^{(s,d)} - \sum_{\{j:(j,i)\in\mathcal{A}\}} f_{j,i}^{(s,d)} = \begin{cases} -r_{s,d} & \text{if } i = d, \\ r_{s,d} & \text{if } i = s, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{N}. \tag{3}$$

2.4.2 Steiner-Tree-Based Multicast Transmission

Steiner-tree-based multicast transmission with node set $S_t = \{n_{t,0}, n_{t,1}, \cdots, n_{t,|S_t|-1}\}$ is a special combination of $|S_t| - 1$ unicast transmissions. Each unicast flow of them should satisfy the flow conservation constraint. Moreover, there is only one path to route a message for each unicast from source $n_{t,0}$ to one destination $n_{t,i} (1 \le i \le |S_t| - 1)$. The difference between the Steiner-tree-based multicast with node set $S_t = \{n_{t,0}, n_{t,1}, \cdots, n_{t,|S_t|-1}\}$ and the $|S_t| - 1$ unicasts from node $n_{t,0}$ to each node in $\{n_{t,1}, \cdots, n_{t,|S_t|-1}\}$ is that, in the former, the consumed resource of each arc (i, j) is the maximum one of $g_{i,j}^{(n_{t,0},n_{t,1})}, \cdots, g_{i,j}^{(n_{t,0},n_{t,|S_t|-1})}$, whereas in the latter, the consumed resource of each arc (i, j) is the sum of $f_{i,j}^{(n_{t,0},n_{t,1})}, \cdots, f_{i,j}^{(n_{t,0},n_{t,|S_t|-1})}$. It is this difference that induces the effectiveness of Steiner-tree-based multicast in utilizing the available communication resource. The flow constraint of Steiner-tree-based multicast transmissions can be expressed as

$$\sum_{\{j:(i,j)\in\mathcal{A}\}} g_{i,j}^{(n_{t,0},n_{t,l})} - \sum_{\{j:(j,i)\in\mathcal{A}\}} g_{j,i}^{(n_{t,0},n_{t,l})} = \begin{cases} -R_t & \text{if } i = n_{t,l}, \\ R_t & \text{if } i = n_{t,0}, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{N}, l \in \{1, \cdots, |S_t| - 1\}, \tag{4a}$$

$$\sum_{\{j:(i,j)\in\mathcal{A}\}} F\left(g_{i,j}^{(n_{t,0},n_{t,l})}\right) \le 1, \ \forall i \in \mathcal{N}, \tag{4b}$$

$$\sum_{\{j:(j,i)\in\mathcal{A}\}} F\left(g_{j,i}^{(n_{t,0},n_{t,l})}\right) \le 1, \ \forall i \in \mathcal{N}. \tag{4c}$$

2.4.3 Network-Coding-Based Multicast Transmission

When network coding is used, the problem of establishing a multicast connection with node set $S_t = \{n_{t,0}, n_{t,1}, \cdots, n_{t,|S_t|-1}\}$ and traffic rate R_t equates to two essentially decoupled problems: One is determining the subgraph in the current network (that is, determining how much flow to put on each link), and the other is determining the code to use over that subgraph (that is, specifying how to encode packets together at each related node) [19]. The necessary and sufficient condition for the feasibility of a subgraph is shown in (5) [19]. Different feasible subgraphs may have different resource consumptions. Once we select a feasible subgraph, any feasible code can be used to implement this multicast connection:

$$\sum_{\{j:(i,j)\in\mathcal{A}\}} g_{i,j}^{(n_{t,0},n_{t,l})} - \sum_{\{j:(j,i)\in\mathcal{A}\}} g_{j,i}^{(n_{t,0},n_{t,l})} = \begin{cases} -R_t & \text{if } i = n_{t,l}, \\ R_t & \text{if } i = n_{t,0}, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{N}, l \in \{1, \cdots, |S_t| - 1\}. \tag{5}$$

Such multicast is another special combination of $|S_t| - 1$ unicasts. Each unicast flow of them satisfies the flow conservation constraint, as shown in (5). However, different from the case in Steiner-tree-based multicast, there can be multiple paths to route a message simultaneously for each unicast from source $n_{t,0}$ to one destination (that is, no constraints (4b) and (4c)). For example, in Fig. 1, paths $s \to a \to t_1$ and $s \to b \to c \to d \to t_1$ are from s to t_1, and paths $s \to b \to t_2$ and $s \to a \to c \to d \to t_2$ are from s to t_2. Obviously, like the multipath routing in [20], network-coding-based multicast routing can also balance the network load. The optimal routing in [20] applies a multipath routing technique for each unicast connection to achieve a system-optimal objective, but it brings no benefit in terms of resource consumption from the perspective of each unicast. For a multicast connection, the purpose of applying network-coding-based routing instead of Steiner-tree-based routing is to achieve the user-optimal routing, which can significantly reduce the bandwidth consumption of each connection [5] and thus reduce the overall resource consumption in a network. The same as the case in Steiner-tree-based multicast, the consumed bandwidth of each arc (i, j) is the maximum one of $g_{i,j}^{(n_{t,0},n_{t,1})}, \cdots, g_{i,j}^{(n_{t,0},n_{t,|S_t|-2})}$ and $g_{i,j}^{(n_{t,0},n_{t,|S_t|-1})}$, instead of the sum of them. Therefore, Steiner-tree-based multicast is a special case of network-coding-based multicast. The network-coding-based minimum-cost multicast is at least as effective as Steiner-tree-based multicast and generally more effective than Steiner-tree-based multicast [5].

2.5 Network-Coding-Based Minimum-Cost Multicast

Denote by $a_{i,j}$ the cost per unit flow on arc (i, j). In a network-coding-based network represented by $\mathcal{G}(\mathcal{N}, \mathcal{A})$, the problem of constructing a single minimum-cost multicast connection with node set $S_t = \{n_{t,0}, n_{t,1}, \cdots, n_{t,|S_t|-1}\}$ can be formulated as follows [5], [19], [21], [22]:

$$\begin{aligned}&\text{Minimize :} \quad \sum_{(i,j)\in\mathcal{A}} a_{i,j}\cdot z_{i,j}\\ &\text{Subject to :}\\ &z_{i,j}\geq g_{i,j}^{(n_{t,0},n_{t,l})}, \forall (i,j)\in\mathcal{A}, l\in\{1,\cdots,|S_t|-1\},\end{aligned} \tag{6}$$

$$\sum_{\{j:(i,j)\in\mathcal{A}\}} g_{i,j}^{(n_{t,0},n_{t,l})} - \sum_{\{j:(j,i)\in\mathcal{A}\}} g_{j,i}^{(n_{t,0},n_{t,l})} = \begin{cases} -R_t & \text{if } i=n_{t,l},\\ R_t & \text{if } i=n_{t,0},\\ 0 & \text{otherwise,}\end{cases} \tag{7}$$
$$\forall i\in\mathcal{N}, l\in\{1,\cdots,|S_t|-1\},$$

$$C_{i,j}\geq g_{i,j}^{(n_{t,0},n_{t,l})}\geq 0,\ \forall (i,j)\in\mathcal{A}, l\in\{1,\cdots,|S_t|-1\}. \tag{8}$$

This is a linear programming problem with polynomial-time algorithms to obtain the optimal solution. In our topology design algorithms, we regard distance $d_{i,j}$ as $a_{i,j}$ and construct the minimum-cost multicast connection for each multicast requirement.

2.6 Link Utilization Constraint

We assume that communication lines are bidirectional (that is, signals can be carried in both directions simultaneously). This assumption is true in most practical cases. In a network $\mathcal{G}(\mathcal{N},\mathcal{A})$, the total amount of unicast flows and multicast flows on an arc (i,j) should be less than or equal to $C_{i,j}$, the capacity assigned to link $\{i,j\}$. This constraint can be expressed as

$$\sum_{i_1=1}^{N}\sum_{\substack{i_2=1\\ i_2\neq i_1}}^{N} f_{i,j}^{(i_1,i_2)} + \sum_{t=1}^{M}\max_{l\in\{1,\cdots,|S_t|-1\}} g_{i,j}^{(n_{t,0},n_{t,l})}\leq C_{i,j}, \tag{9}$$
$$\forall (i,j)\in\mathcal{A}.$$

The first term on the left-hand side of (9) is the total amount of unicast traffic on arc (i,j) and the second term is the total amount of network-coding-based multicast traffic on arc (i,j). Note that, as mentioned previously, for the tth multicast, the amount of traffic on arc (i,j) is the maximum one of $|S_t|-1$ unicast flows, that is, $\max_{l\in\{1,\cdots,|S_t|-1\}} g_{i,j}^{(n_{t,0},n_{t,l})}$, instead of the sum of $|S_t|-1$ unicast flows on arc (i,j).

2.7 Delay Requirement

It is necessary to keep the average end-to-end packet (AEEP) delay (a network-wide metric) within an admissible value. In most available literature, an M/M/1 queuing model based on Kleinrock's independence assumptions is adopted to calculate the average packet delay on each network link. Based on this model, the AEEP delay can be expressed as

$$T=\frac{1}{\gamma}\sum_{(i,j)\in\mathcal{A}}\frac{f_{i,j}}{C_{i,j}-f_{i,j}}, \tag{10}$$

where γ is the total arrival rate into the network in packets per second and $f_{i,j}$ and $C_{i,j}$ are the total traffic rate on arc (i,j) and the capacity of arc (i,j) in bits per second [13], [20].

However, it is inappropriate to still apply this model to current high-speed multiservice networks. One reason is that (10) considers neither propagation delay nor nodal processing delay, both of which are very important in high-speed networks where it is unrealistic to neglect them. Another important reason is that high-speed networks are capable of carrying many types of services such as voice, data, and video, whose corresponding packets are probably separated in different queues with different priorities, rather than one queue.

The appropriate delay model for current and future networks is related with the specific packet scheduling scheme adopted, and it is far more complex than the traditional one. It is not desirable to embed a burdensome analysis of delay in the complex topology design. In addition, it is possible that in a network meeting the AEEP delay constraint, most requirements have small AEEP delays and some requirements have large AEEP delays. It is preferable to create a more balanced design. The more balanced design is also better able to withstand variations in the requirement level and distribution.

A delay-balanced design can be obtained by limiting the utilization of each arc separately [14]. In our topology design problem, a limit (or threshold) is imposed on the utilization of each arc to control packet delay. Denote the maximum permitted utilization of each arc by $e_{\max}$. Regretfully, we cannot get an explicit relationship between parameter e_{max} and the AEEP delay. Nevertheless, some researchers have studied the effect of link utilization on the delay performance [23], [24], [25] and obtained some results. For example, for a link loaded with TCP traffic composed by many TCP connections, when the global offered load increases above 80 percent, the performance of each single connection decreases very quickly [23]. The results of these papers can provide us some general guidelines about the value specification of parameter e_{max}.

This constraint on arc utilization is more stringent than previous link utilization constraints.

2.8 Formulation

Now, the topology design problem we consider can be formulated as follows:

Given

1. node number N and distance matrix $(d_{i,j})_{N\times N}$,
2. unicast requirement matrix $(r_{i,j})_{N\times N}$,
3. the node set $\{n_{i,0},n_{i,1},\cdots,n_{i,|S_i|-1}\}$ and the traffic rate R_i of the ith multicast requirement $(i=1,2,\cdots,M)$,
4. capacities $C_1,\cdots,C_K$, fixed costs $f_1,\cdots,f_K$, and costs per unit length $p_1,\cdots,p_K$ of different types of lines,
5. connectivity k, and
6. maximum arc utilization $e_{\max}$,

Minimize

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} D_{i,j}=\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{t=1}^{K} u_{i,j}^{t}(f_t+d_{i,j}\cdot p_t)$$

Over the design variables

$$u_{i,j}^{1},\cdots,u_{i,j}^{K}\in\mathbb{N}: 1\leq i\leq N-1, i+1\leq j\leq N$$
$$f_{i,j}^{(s,d)}\geq 0: 1\leq i,j,s,d\leq N(i\neq j, s\neq d)$$
$$g_{i,j}^{(n_{t,0},n_{t,l})}\geq 0: t\in\{1,\cdots,M\}, l\in\{1,\cdots,|S_t|-1\}, 1\leq i,j\leq N(i\neq j)$$

Subject to

1. $C_{i,j} = u_{i,j}^1 C_1 + \cdots + u_{i,j}^K C_K$, where $u_{i,j}^1, \cdots, u_{i,j}^K \in \{0, 1, 2, \cdots\}$, $C_{j,i} = C_{i,j}, \forall 1 \le i \le N-1, i+1 \le j \le N$.
2. The network reliability requirement

$$\sum_{i \in S} \sum_{j \in \{1,2,\cdots,N\} \setminus (Z \bigcup S)} F(C_{i,j}) \ge 1,$$
$$\forall 1 \le s, d \le N (s \ne d),$$
$$\forall Z \subseteq \mathcal{N} \setminus \{s, d\} \text{ with } |Z| = k-1,$$
$$\forall S \subseteq \mathcal{N} \setminus Z \text{ with } s \in S \text{ and } d \notin S.$$

3.1. Unicast flow conservation constraint

$$\sum_{\substack{1 \le j \le N \\ j \ne i}} f_{i,j}^{(s,d)} - \sum_{\substack{1 \le j \le N \\ j \ne i}} f_{j,i}^{(s,d)} = \begin{cases} -r_{s,d} & \text{if } i = d, \\ r_{s,d} & \text{if } i = s, \\ 0 & \text{otherwise}, \end{cases}$$
$$\forall 1 \le i, s, d \le N (s \ne d).$$

3.2. Multicast flow conservation constraint

$$\sum_{\substack{1 \le j \le N \\ j \ne i}} g_{i,j}^{(n_{t,0}, n_{t,l})} - \sum_{\substack{1 \le j \le N \\ j \ne i}} g_{j,i}^{(n_{t,0}, n_{t,l})} = \begin{cases} -R_t & \text{if } i = n_{t,l}, \\ R_t & \text{if } i = n_{t,0}, \\ 0 & \text{otherwise}, \end{cases}$$
$$\forall t \in \{1, \cdots, M\}, l \in \{1, \cdots, |S_t| - 1\}, 1 \le i \le N.$$

4. Link utilization constraint and delay requirement

$$f_{i,j} = \sum_{i_1=1}^{N} \sum_{\substack{i_2=1 \\ i_2 \ne i_1}}^{N} f_{i,j}^{(i_1,i_2)} + \sum_{t=1}^{M} \max_{l \in \{1,\cdots,|S_t|-1\}} g_{i,j}^{(n_{t,0}, n_{t,l})}$$
$$\le e_{max} \cdot C_{i,j}, \ \forall 1 \le i, j \le N (i \ne j).$$

Compared with traditional topology design problems, this problem has an additional constraint, that is, the flow conservation constraint of network-coding-based multicast transmissions. In addition, because there are multicast transmissions, when compared with conventional problems, constraint (4) has an additional term reflecting the characteristic of network coding.

Lemma 1. *The topology design problem of survivable (that is, k-node connected) unicast networks is NP-hard.*

Proof. This topology design problem is NP-hard even when the traffic requirement $r_{i,j}$ $(i, j \in V \text{ and } i \ne j)$ is very small, such that the smallest capacity C_1 is enough for each link to be assigned, because it contains some known NP-hard problems such as the traveling salesman problem and the connectivity augmentation problem as special cases [11], [12]. □

Theorem 1. *The topology design problem of survivable network-coding-based multicast networks is NP-hard.*

Proof. This new topology design problem of survivable network-coding-based multicast networks contains the traditional unicast-oriented design problem as a special case and, thus, is also NP-hard. □

No polynomial-time algorithms are available to obtain the optimal solution of an NP-hard optimization problem. It is necessary to develop heuristic algorithms to deal with it.

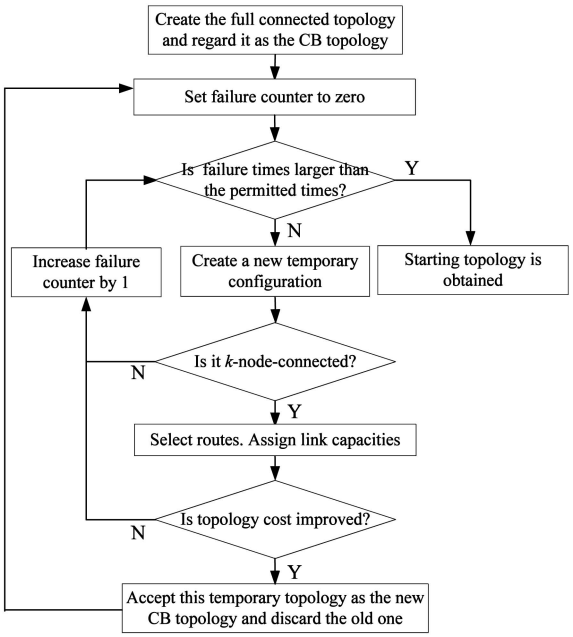

Fig. 2. Flowchart of the starting topology generation in the LDE algorithm.

3 Heuristic Algorithms

In this section, we will introduce two heuristic algorithms, the LDE algorithm and the LAE algorithm, for this topology design problem.

These two proposed algorithms are both composed of two phases, starting topology generation and local optimization process. In the first phase of the LDE algorithm, by deleting links one by one from the fully connected topology until no link can be deleted anymore, a k-node-connected starting topology with relatively low cost is generated. In the first phase of LAE algorithm, by adding links one by one from the original topology with no link until no one more link is needed anymore, a k-node-connected starting topology with a relatively low cost is generated. In the second phase of both algorithms, link exchange is iteratively performed to locally improve the starting topology step by step.

For simplicity, we first consider the case that only a line can be assigned to each link $\{i, j\}$, that is, $C_{i,j} \in \{0, C_1, \cdots, C_K\}$. Then, these two algorithms will be extended to the general case that several communication lines can be assigned on each link.

3.1 Link Deletion and Exchange Algorithm

3.1.1 Starting Topology Generation

The objective of this phase is to generate a k-node-connected topology whose cost is relatively low. The flowchart of this phase is shown in Fig. 2.

First, create the fully connected topology and regard it as the current best (CB) topology. Then, obtain a temporary configuration by deleting a particular link in the current configuration. If this temporary configuration satisfies some particular conditions, it means that, based on this temporary configuration, a new feasible topology with a lower cost can be obtained. Accept this new feasible topology as the new CB topology, discard the old one, and set parameter t,

which is a counter parameter used to count the continuous failure times, back to zero. If this temporary configuration does not satisfy all those conditions, discard it and increase t by one. If the value of t exceeds a given value $t_{\max}$, terminate the algorithm, and the CB topology is the final topology of this phase. Otherwise, obtain another temporary configuration and test it. In this way, link deletion operation is conducted repeatedly until no appropriate link can be deleted any more.

Define an efficiency metric $m_{i,j}$ on each link $\{i,j\}$ by $m_{i,j} = D_{i,j}/(f_{i,j} + f_{j,i})$.

This process consists of the following detailed steps:

1. Index N nodes from node 1 to N randomly and create the fully connected configuration. Then, select the route for each requirement and allocate link capacities. Regard the resulting topology as the CB topology.

   Routing and capacity allocation procedure. For each unicast requirement, select the shortest distance path between the source node and the determination node as its route, and for each multicast requirement, select the route obtained by the network-coding-based minimum-cost multicast algorithm as its route.[1]

   For each link $\{i,j\}$, assign to it the smallest capacity in the set $\{0, C_1, \cdots, C_K\}$ that is greater than or equal to

$$\frac{1}{e_{max}}\left(\sum_{i_1=1}^{N}\sum_{\substack{i_2=1\\ i_2\neq i_1}}^{N} f_{i,j}^{(i_1,i_2)} + \sum_{t=1}^{M}\max_{l\in\{1,\cdots,|S_t|-1\}} g_{i,j}^{(n_{t,0},n_{t,l})}\right) \quad (11)$$

   and

$$\frac{1}{e_{max}}\left(\sum_{i_1=1}^{N}\sum_{\substack{i_2=1\\ i_2\neq i_1}}^{N} f_{j,i}^{(i_1,i_2)} + \sum_{t=1}^{M}\max_{l\in\{1,\cdots,|S_t|-1\}} g_{j,i}^{(n_{t,0},n_{t,l})}\right). \quad (12)$$

2. Set counter parameter t to zero and initialize E, which consists of the candidate links to delete, to the set consisting of all links in the CB topology.
3. Check whether the value of t is larger than $t_{\max} = \lceil N \cdot k/2 \rceil$. If it is, go to step 7.
4. From E, select the link l whose efficiency metric value is largest. Obtain a temporary configuration by removing link l from the current configuration.

   Test whether this temporary configuration is k-node connected. If it is not, discard it, increase t by one, and remove link l from candidate link set E. Then, go back to step 3.
5. Assign routes again only for those unicast requirements and multicast requirements whose routes pass through link l in the CB topology.
6. Calculate the total cost of all links. If the topology cost is improved (that is, lower), accept this temporary topology as the CB topology. Then, go back to step 2.

   If it is not, discard the temporary configuration, increase t by one, and remove link l from the candidate link set E. Then, go back to step 3.
7. Exit and return the CB topology.

In step 3, the reason why we let $t_{\max}$ equal $\lceil N \cdot k/2 \rceil$ is that each CB topology that is k-node connected has at least $\lceil N \cdot k/2 \rceil$ links.

3.1.2 Local Optimization Process

In this phase, the starting topology obtained in the first phase will be improved by exchanging two links iteratively.

Given two links, there are several possible cases of link exchange. If these two links are adjacent, that is, they have a common node, after exchanging these two links, the configuration remains unchanged. If these two links are not adjacent, there are two possible exchange schemes. In more detail, given links $\{A,B\}$ and $\{C,D\}$, where node A, B, C, and D are different from each other, we can exchange them to new links $\{A,C\}$ and $\{B,D\}$ or to new links $\{A,D\}$ and $\{B,C\}$. If one old link and one new link are the same, we regard them as one link. Maybe one or both of these two exchange schemes will cause a new feasible topology with a lower cost or maybe neither of them will cause a new feasible topology with a lower cost.

The main idea of this process is as follows: For the CB topology, select two candidate links to exchange. If a feasible topology with a lower cost can be obtained by link exchange, accept this topology as the CB topology and continue to improve this new CB topology by link exchange. If no feasible topology with a lower cost can be obtained by link exchange, continue to select another two candidate links to test. If, finally, all possible link pairs have been tried and no better topology can be obtained, terminate the algorithm, and the CB topology is the final topology.

The order of link pairs for testing in the CB topology is determined by the following rule: Assume that there are l links in the CB topology. First, index these l links from 1 to l such that, if $i < j$, the efficiency metric value of link i is larger than that of link j. For each link pair (link i, link j), define a metric $S = i + j$. Then, sort all link pairs according to their values of metric S in ascending order. As for the order of those link pairs with the same metric value, sort them according to the smaller index in each link pair. For example, for link pairs (link 1, link 4) and (link 2, link 3), their values of metric S are both 5. The smaller index in (link 1, link 4) is 1, and the smaller index in (link 2, link 3) is 2. Thus, (link 1, link 4) ranks ahead of (link 2, link 3). The order of link pairs is shown as follows: (link 1, link 2), (link 1, link 3), (link 1, link 4), (link 2, link 3), (link 1, link 5), (link 2, link 4), ...

The flowchart of the local optimization process is shown in Fig. 3. It consists of the following steps:

1. Set counter parameter t, which is used to count the continuous failure times, to zero. For the CB topology, obtain the link pair order according to the rule described above.
2. Check whether the value of t is larger than $t_{\max} = \binom{l}{2}$. If it is, go to step 5.
3. Select the link pair (link i, link j) that has not been tested according to the link pair order.

1. The minimum-cost multicast route here is obtained by relaxing (discarding) the constraints in (8), that is, each link capacity is considered as infinite.

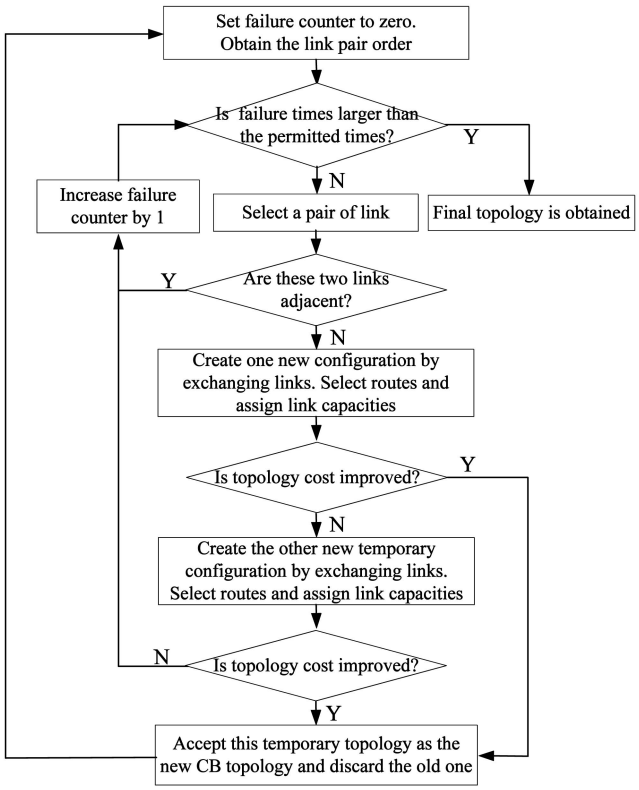

Fig. 3. Flowchart of the local optimization process.

4. If link i and link j are adjacent, increase t by one and go back to step 2. If link i and link j are not adjacent, there are two possible exchange schemes. Pick an arbitrary one and conduct the following test first. If this exchange scheme cannot prompt a better topology, then select the other exchange scheme and also conduct following test.

   Feasibility test. After link exchange, we get a new configuration. Determine if it is k-node connected. If it is not, then this link exchange cannot induce a feasible topology. Otherwise, select the route for each requirement, allocate link capacities, and then calculate the total cost of this new topology. If this total cost is lower than that of the CB topology, discard the CB topology, regard this new topology as the new CB topology, and go back to step 1.

   If both link exchange schemes cannot prompt a better topology, increase t by one and go back to step 2.
5. Exit and return the CB topology.

3.2 Link Addition and Exchange (LAE) Algorithm

This algorithm also consists of two phases, starting topology generation and local optimization process, and the second phase is the same as that of the LDE algorithm. Hence, here, we only describe the first phase.

3.2.1 Starting Topology Generation

The main idea of this phase is that we first generate a k-node-connected configuration that has the potential to be a low-cost topology and then build a topology based on this configuration.

This phase consists of the following detailed steps:

1. Index N nodes from 1 to N randomly.
2. Determine the node with the smallest degree. Call this node X. If there are several candidate nodes, select the one with the smallest index. Determine the node with the smallest degree that is not already connected to X. Call this node Y. If there are several candidate nodes, select the one that is nearest to X. Add the link $\{X, Y\}$.
3. Repeat step 2 until each node's degree is at least k.
4. Check whether the current configuration is k-node connected. If it is, go to step 6.
5. Check whether the connectivity of the current configuration can be increased (by one) by only adding one link. If it can be, add the shortest link whose addition can increase the connectivity. Otherwise, discard the current configuration and go back to step 1.

   Repeat the above operation until the current configuration is k-node connected or until the connectivity of the current configuration cannot be increased by one by only adding one link.
6. Then, select the route for each requirement and allocate link capacities.
7. Exit and return the CB topology.

In step 5, if more than one link must be added to increase the connectivity, the rule is quite complex to determine which links are appropriate to add to guarantee that the resulting topology has a low cost [26].

3.3 Complexity Analysis

The running time for testing k-node connectivity is $O(k^2N|E|)$, where E is the link set [10].

The complexity of routing for all unicast requirements is $O(N^3)$ [10]. There are M multicast requirements. For each one of them, the simplex method[2] is adopted to obtain the minimum-cost route. The expected complexity of the simplex method is $O(m^2n)$, where m is the number of constraint equations and n is the number of variables in the linear programming problem [27]. Then, the expected complexity to build a multicast route is $O(|E|^3|S|^3)$, where S is the multicast node set. Routing for M multicast requirements takes time $O(M|E|^3|S|^3)$.

According to (11) and (12), it is easy to know that the allocating capacities for $|E|$ links takes time $O(N^2|E|)$. According to (1), the cost calculation of a topology takes time $O(K \cdot N^2)$.

3.3.1 Computational Complexity of the LDE Algorithm

During the first phase, for each new temporary configuration, either only connectivity testing is done or all the operations listed in Table 2 are done. Among these operations, multicast routing is the most time consuming. In the worst case, for each CB topology with $|E|$ links, $|E|$ temporary configurations are all tested until the $|E|$th test before a better topology is obtained. However, our simulation shows that at almost all iterations (other than the last

2. There exist polynomial algorithms for linear programming. Although the simplex method takes exponential time in the worst case, we adopt it because of its remarkable efficiency in practice.

TABLE 2
Runtime of Different Operations

	Connectivity testing	Unicast routing	Multicast routing	Capacity allocation	Cost calculation
Complexity	$O(k^2N\|E\|)$	$O(N^3)$	$O(M\|E\|^3\|S\|^3)$	$O(N^2\|E\|)$	$O(K \cdot N^2)$

several iterations), after only testing several temporary configurations, a better topology can be obtained, which is far better than the worst case. Thus, it is more useful to analyze the average-case complexity. The runtime of the first phase is

$$\begin{aligned} T_1 &= O\left(M\left(\frac{N^2-N}{2}\right)^3|S|^3 + M\left(\frac{N^2-N}{2}-1\right)^3|S|^3 + \right. \\ &\quad \left. \cdots + M\left(\frac{kN}{2}\right)^3|S|^3\right) \\ &= O(M|S|^3N^8). \end{aligned}$$

The topology obtained from the first phase has around $k \cdot N/2$ links and thus has around $O(k^2N^2)$ different link pairs. During the second phase, the topology will be improved repeatedly. According to our simulation, the times of improving the CB topology is $O(N)$. The runtime of the second phase is

$$T_2 = O(M|E|^3|S|^3) \cdot O(k^2N^2) \cdot O(N) = O(k^5M|S|^3N^6).$$

Overall, the runtime of the LDE algorithm is $O(M|S|^3N^6(N^2+k^5))$.

3.3.2 Computational Complexity of the LAE Algorithm

In the first phase, it takes time $O(kN^3)$ to construct a configuration in which each node's degree is at least k, and according to our simulation experience, we run steps 1 to 5 $O(N)$ times to get a k-node-connected configuration. In step 6, routing and capacity allocation take time $O(Mk^3N^3|S|^3)$. Hence, the overall runtime of the first phase is $O(kN^4 + Mk^3N^3|S|^3)$, which is far lower than the runtime of the second phase.

The overall runtime of the LAE algorithm is $O(k^5M|S|^3N^6)$.

One potential way for reducing the complexity is by adopting a suboptimal routing having low complexity, instead of the minimum-cost routing, to build routes for multicast requirements.

3.4 General Case of the Link Capacity Assignment

If more than one line can be assigned to one link, the only difference between new algorithms and the above algorithms is in capacity assignment. The new capacity assignment is to determine the quantity of each link type. Here, we explore this problem in brief.

The capacity assignment problem of link $\{i,j\}$ can be formulated as follows:

$$\min \sum_{t=1}^{K} u_{i,j}^t(p_t + f_t/d_{i,j}), \quad \text{where} \quad u_{i,j}^t = 0,1,2,\cdots,$$

subject to

$$\sum_{t=1}^{K} u_{i,j}^t C_t \geq f_{i,j}/e_{\max}.$$

This problem can be iteratively solved by dynamic programming methods [28].

4 Simulation Results and Analysis

In this section, first, we will compare the LDE algorithm with the LAE algorithm and determine which one is better according to simulation results. Then, we will evaluate the effectiveness of the better algorithm by comparing it with the exhaustive search (ES) method in small-size networks. Finally, the benefit brought by the network coding technique in topology design is shown by comparing the coding-based design with the unicast-oriented design.

4.1 Simulation Parameter Settings

Information about the available types of communication lines is shown in Table 3. The fixed costs are set to appropriate values so that, in the resulting topologies, the total fixed cost accounts for around 25 percent of the total cost. Unit length costs of different types of lines follow the principle of scale of economy. In our simulations, $e_{\max}$ is set to 0.85, and unless otherwise mentioned, we consider designing three-node-connected topologies, that is, k equals three.

In practice, the amount of traffic from node i to j is different but generally not very different from the amount of traffic from node j to i [29]. Hence, in our tests, we set the unicast rate in the following way: Unicast requirement rate $r_{i,j}(i<j)$ is selected uniformly in the interval $[r_{\min}, r_{\max}]$ Mbps, and unicast requirement rate $r_{j,i}$ is selected uniformly in the interval $[0.6r_{i,j}, 1.4r_{i,j}]$ Mbps.

In a network with N nodes, there are a total of $N(C_{N-1}^2 + \cdots + C_{N-1}^{N-1})$ possible multicast requirements. However, it is not difficult to imagine that, in practice, most of them are with low rates. It is impractical and not quite necessary to consider all multicast requirements specially. It is practical that, at the stage of traffic requirement estimation, only those multicast requirements with moderate or high rates are considered separately and the traffic of low-rate multicast requirements is considered as unicast traffic. In our tests, there are $3N$ multicast

TABLE 3
Available Capacity Options and Costs

Capacity (Mbps)	Variable cost (unit cost/unit length)	Fixed cost (unit cost)
100	1	80
300	2	100
600	3.5	120
1000	5	160
1500	7	200

TABLE 4
Node Locations

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
X	344	168	154	10	168	158	195	310	315	393	277	292	173	474	190	468
Y	224	139	262	41	287	130	127	196	42	104	193	173	228	239	199	179

requirements and the number of sinks of each multicast is selected uniformly in the integer interval $[2, N-1]$. Each multicast requirement rate is selected uniformly in the interval $[R_{\min}, R_{\max}]$ Mbps. The parameters $r_{\min}$, $r_{\max}$, $R_{\min}$, and $R_{\max}$ are used to adjust the unicast traffic amount and the multicast traffic amount.

4.2 Comparison of Two Heuristic Algorithms

The topology cost resulting from an algorithm depends on input parameter values and the performance of this algorithm. The *workload* (that is, the total amount of traffic originating from all nodes) and the ratio of multicast traffic amount[3] to the total network traffic amount, somewhat vaguely called *traffic ratio*, are two important input parameters closely related to topology cost. The larger the workload is, the higher the resulting topology cost is. Given a workload, the larger the traffic ratio is, the lower the resulting topology cost is if the topology design algorithm takes advantage of the multicast characteristic.

To evaluate the performance of our algorithms, we consider a set of 16 nodes whose positions are randomly selected in a scale of 500×300 unit distance. Table 4 shows the node locations represented by the set of Cartesian coordinates X and Y. Based on these 16 nodes, we investigate the performance of the two proposed algorithms under different workloads and different traffic ratios.

If, for any two nodes i and j, the amount of traffic from i to j equals that from j to i, we say the traffic is symmetric; otherwise, the traffic is asymmetric. Let us illustrate the effect of the symmetry of traffic on the topology cost through an example about the traffic in a communication line. In one case, 70-Mbps traffic is transmitted in one direction and 70-Mbps traffic is in the other direction. In another case, 20-Mbps traffic is transmitted in one direction and 120-Mbps traffic is in the other direction. Although, in both cases, the total loads in this line are equal, a capacity of 100 Mbps is enough for it in the first case and a capacity of 300 Mbps is needed for it in the second case. Thus, if the traffic in the network is highly asymmetric, the cost of the resulting topology is higher than that resulting from the same amount of relatively symmetric traffic.

4.2.1 Comparison under Different Workloads

First, we investigate the performances of two algorithms under different workloads with a traffic ratio of 40 percent. For each workload, we obtain the average topology cost of a number of cases with different spatial distributions of traffic among 16 nodes. Fig. 4 shows the average topology costs under different workloads of the LDE and LAE algorithms. For each algorithm, the average topology cost increases approximately linearly with increasing workload. This is very explicit, since more traffic will consume more capacity in the resulting topology. In addition, the principle of scale of economy about the line cost is demonstrated here. Take the LAE algorithm as an example. When the workload increases from 3,000 to 7,000 Mbps, the average topology cost only increases to around 1.5 times.

It is interesting to note in Fig. 4 that, when the workload is not very high (for example, below 6,500 Mbps), the LAE algorithm always performs better than the LDE algorithm. When the workload is high (for example, above 7,000 Mbps), however, the LDE algorithm actually outperforms the LAE algorithm. This topology cost crossover observation in Fig. 4 is actually due to a similar crossover in the network-wide average link (NAL) utilization of the two algorithms. As the workload increases from 3,000 to 7,000 Mbps, the NAL utilization of the LDE algorithm increases from 50.6 percent to 61.5 percent, whereas the NAL utilization of the LAE algorithm grows from 56.3 percent to 60.7 percent.[4] It is notable that the topology cost is heavily related to the NAL utilization, since a low NAL utilization usually results in a high-cost topology. The NAL utilization crossover of the two algorithms can be explained by their differences in the number of links of the final topology designs. The number of network links that resulted from the LDE algorithm is mainly distributed in the interval [26, 30], whereas the numbers of network links that resulted from the LAE algorithm is usually 24 or 25.[5] When the workload is low (for example, 3,000 Mbps), links in LDE-based topology designs usually carry a lower amount of traffic and, thus, have lower link utilizations than those in the LAE-based topology designs (note that the smallest capacity that can be allocated is 100 Mbps) because the topologies obtained from the LDE algorithm usually have more links than those from the LAE algorithm to support the same workload. When the workload is high, however, we can actually benefit from the topologies that have more links. For a given multicast connection, the coding-based minimum-cost route generally consumes lower bandwidth and has a better load-balance capability in topologies with more links. A more uniform distribution of multicast traffic can actually relieve the negative effect caused by the traffic asymmetry and, thus, increase the NAL utilization.

3. The traffic amount of a multicast transmission with transmission rate R and t receivers is considered as $R \cdot t$.

4. The reason the NAL utilization increases as the workload increases is that, when the workload is low, many arcs carry a small amount of traffic and are underutilized (note that the smallest capacity that can be allocated is 100 Mbps), but as the workload increases, the traffic amount over each arc will grow, and consequently, the NAL utilization will increase for both the LDE and the LAE algorithms. In addition, the NAL utilization is not very high here due to the asymmetric traffic distribution and the imposed constraint on link utilization.

5. The link number of the resulting topology depends on the link deletion process in the LDE algorithm and on the link addition process in the LAE algorithm.

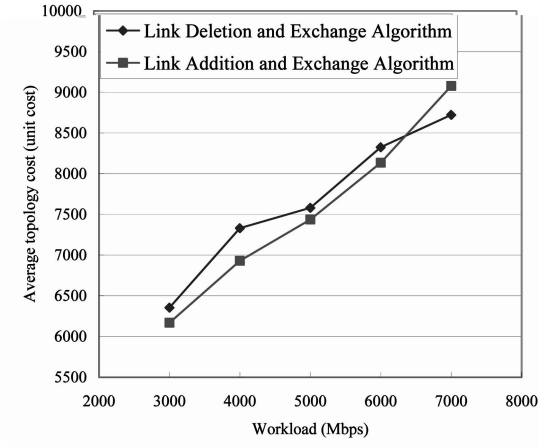

Fig. 4. Average topology costs versus workloads with $\text{traffic ratio} = 40$ percent.

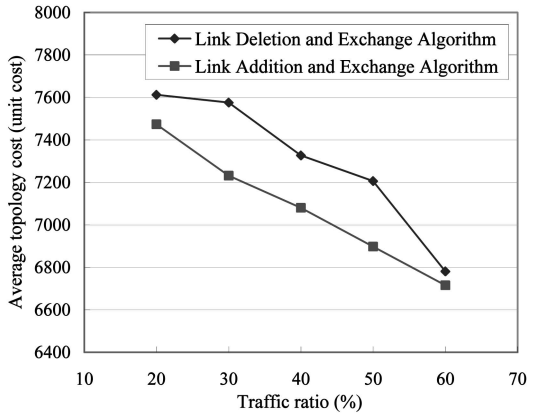

Fig. 5. Average topology costs versus traffic ratios with a moderate workload.

4.2.2 Comparison under Different Traffic Ratios

Now, we investigate the performances of the two algorithms under the same workload and different traffic ratios. Note that it is often not practical for the workload of a network design to be very high; thus, the evaluation is performed under a moderate workload. For each traffic ratio, we obtain the average topology cost of a number of cases with different spatial distributions of traffic among 16 nodes. Fig. 5 shows the average topology costs of different ratios. For each algorithm, the average cost approximately linearly decreases with the increase of the traffic ratio. It is easy to understand such a tendency, since a certain amount of multicast traffic will consume less resource than that consumed by the same amount of unicast traffic. Therefore, for a given workload, the higher the percentage that multicast traffic accounts for, the less total capacity the resulting topology needs.

In Fig. 5, we can see that the average cost does not decrease fast with the increase of the traffic ratio, partially because of the asymmetric traffic pattern we used for the test. Because the overall computational burden of all simulations is heavy, as mentioned previously, there are only $3N$ multicast requirements in the topology design simulations we conducted. However, we conjecture that, in practice, there are at least $O(N^2)$ multicast requirements with moderate or high rates, and multicast traffic is relatively uniformly distributed among N nodes. If this is true, the average cost will decrease with the increasing traffic ratio at a faster rate than that shown in Fig. 5.

Compared with the LAE algorithm, the average cost of the LDE algorithm increases by 1.8 percent, 4.8 percent, 3.5 percent, 4.5 percent, and 1.0 percent corresponding to a traffic ratio of 20 percent, 30 percent, 40 percent, 50 percent and 60 percent, respectively.

According to the above comparison results, we conclude that, on the whole, the LAE algorithm performs slightly better than the LDE algorithm. Only the performance of the LAE algorithm will be evaluated below.

4.3 Performance Evaluation

The performance of a topology design algorithm can be evaluated through a comparison with available good algorithms on the same problem or by gauging the gap between the topology cost obtained by this algorithm and the lower bound on the cost of the optimal topology [13]. Regretfully, there are no available heuristic algorithms used to design NCM network topologies, and lower bounds are only known for simple cases even for the unicast-oriented topology design problem [28], not to mention the NCM network topology design problem. The approach we take is to compare the LAE algorithm with the ES method.

However, it is impossible to obtain the optimal topology by the ES method even for five-node cases. Here, we briefly deal with the complexity of the ES method for five-node cases. For five-node cases, if there are five types of lines available, there are $6^{N(N-1)/2} \approx 6.0 \times 10^7$ possible topologies to be tested, and for each topology, we should confirm whether it is k-node connected and whether it is with a lower cost. If these two conditions are both satisfied, then we should try all possible flow assignments to confirm whether all requirements can be accommodated simultaneously in this topology. Since the flow from the source to the determination can be split (or divided) and transmitted over multiple paths simultaneously, there are a large number of possible flow assignments.

Hence, we use five four-node cases with different parameter values for the test, and the objective is to obtain two-node-connected low-cost topologies. We make a reasonable assumption that, when the traffic from one node to another is split and transmitted over L paths, the traffic amount on each path should be the times of a basic traffic amount, not an arbitrary amount.

As is shown in Table 5, the LAE algorithm performs almost as good as the ES method in four-node cases. In each case, the difference between the solution cost of the link addition algorithm and the optimal solution cost is typically less than 16 percent. This degree of accuracy is deemed adequate for most topology designs, especially considering that traffic requirements cannot be predicted with much accuracy before network implementation or tend to change during the life of the network. Therefore, we conclude that the link addition algorithm is very effective in designing network-coding-based multicast networks.

4.4 Benefit of Network Coding

When designing the topology of a NCM network, how much can we gain in terms of topology cost by separating

TABLE 5
Comparison between the LAE Algorithm and the ES Method

	Case 1	Case 2	Case 3	Case 4	Case 5
Topology cost (LAE algorithm)	2035.53	2034.46	1788.07	1614.37	1888.22
Topology cost (ES method)	1899.28	1819.78	1727.67	1442.91	1635.29
Cost gap	7.17%	11.80%	3.50%	11.88%	15.47%

TABLE 6
Comparison between MENTOR and ULAE Algorithms

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9
Topology cost (MENTOR)	10359.9	9203.7	10030.3	8480.2	9312.9	8064.59	8275.84	9785.73	6957.69
Topology cost (ULAE)	10505.4	9677.3	9925.9	8468.9	9469.2	7952.4	8283.8	9589.6	6821.8
Cost ratio	0.986	0.951	1.011	1.001	0.983	1.014	0.999	1.020	1.020

multicast requirements from unicast requirements and taking advantage of the characteristic of multicast in topology design algorithms? Furthermore, how much can we gain further if the network coding technique is used to support multicast transmissions?

To answer the above question, we investigate the topology cost difference between the following three cases: In the first case, each multicast requirement is treated as multiple unicast requirements. In the second case, multicast requirements are considered separately from unicast requirements, and the Steiner tree algorithm is used to build multicast routes. In the third case, multicast requirements are considered separately, and the network-coding-based minimum-cost multicast algorithm is used to build multicast routes.

For the first case, conventional unicast-oriented algorithms can be used to design topologies. Unfortunately, no well-known conventional algorithm available deals with exactly the same design problem as ours.[6] One good algorithm used for almost the same design problem as ours is the well-known MENTOR algorithm. The difference is that the problem this algorithm deals with does not include the reliability requirement, whereas the problem we consider includes it. In addition, as far as we know, no well-known algorithm is available for the second case.

The LAE algorithm can be used to design topologies for the first case, like conventional algorithms, by removing the routing procedure for multicast requirements and transforming multicast requirements to unicast requirements. For simplicity, call this revised algorithm the unicast-oriented LAE (ULAE) algorithm. The LAE algorithm can also be used to design topologies for the second case by using Steiner tree algorithms to obtain multicast routes, instead of using the network-coding-based minimum-cost multicast algorithm. Call this revised algorithm the Steiner-tree-based LAE (SLAE) algorithm. In our test, we use the Directed Steiner Tree (DST) approximation algorithm described in [30] to build Steiner trees in the SLAE algorithm. In addition, temporarily call the original LAE algorithm, that is, the network-coding-based one, the network-coding-based LAE (CLAE) algorithm.

6. Topology design problems include a lot of assumptions and requirements. Few well-known algorithms were proposed for exactly the same design problem. For example, some consider the case that there is only one type of line and others consider the case that several types of lines are available. Some consider the reliability requirement and others do not.

4.4.1 Comparison between MENTOR and ULAE Algorithms

Extensive simulations show that, for those cases where three-node-connected topologies are obtained by the MENTOR algorithm, the average cost of topologies obtained by the ULAE algorithm is only 0.28 percent higher than that of the topologies obtained by the MENTOR algorithm.[7] Table 6 shows some comparison results between the MENTOR and ULAE algorithms.

Based on this observation, we can use the ULAE algorithm, the SLAE algorithm, and the CLAE algorithm to investigate the rough gain in terms of topology cost obtained by considering multicast traffic specially and the gain obtained further by using the network coding technique to support multicast.

4.4.2 Comparison between ULAE, SLAE, and CLAE Algorithms

Fig. 6 shows the percent reduction in terms of the average topology cost of the SLAE algorithm and the CLAE algorithm, using the average topology cost of the ULAE algorithm as the base. For the SLAE algorithm, the percent reduction increases slowly with the increase of the traffic ratio. Nevertheless, for the CLAE algorithm, the percent reduction increases rapidly with the increase of the traffic ratio. Take traffic ratio 50 percent as an example. If the network coding technique is used to support multicast transmissions, the average topology cost can be reduced by 16.6 percent, which is far higher than 8.3 percent corresponding to the Steiner-tree-based algorithm. It can be seen in Fig. 6 that network coding can benefit designing topologies, especially when the amount of multicast traffic accounts for a large percentage of the total traffic. We conclude that, when we design multicast network topologies, it is necessary and beneficial to consider multicast traffic specially rather than to treat each multicast as multiple unicasts, and if the technique is adopted, the topology cost can be greatly reduced.

5 Conclusion and Future Work

In this paper, we studied for the first time the challenging topology design problem of network-coding-based multi-

7. The MENTOR algorithm has lower complexity compared to the ULAE algorithm, which is not specially proposed for unicast-oriented topology design.

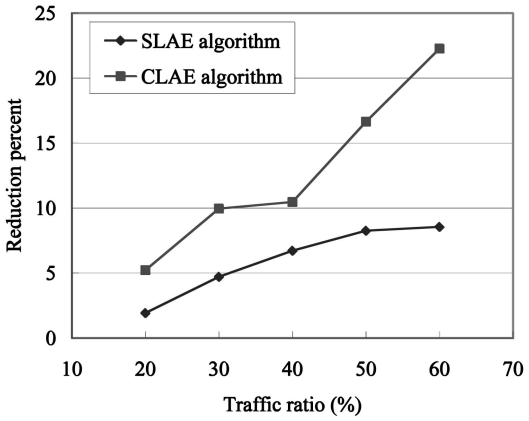

Fig. 6. Percent reduction in terms of the average topology cost of the SLAE algorithm and the CLAE algorithm, using the average topology cost of the ULAE algorithm as the base.

cast networks. Based on the characteristics of multicast and network coding, we formulated this problem as an NP-hard mixed-integer nonlinear programming problem, which is much more complicated than the conventional unicast-oriented topology design problems. Two heuristic algorithms, the LDE algorithm and the LAE algorithm, are proposed for our design problem. Extensive comparisons indicated that, overall, the LAE algorithm performs better than the LDE algorithm, and the LAE algorithm is effective in designing the topologies of network-coding-based multicast networks.

Our results in this paper show that, in comparison with the conventional unicast-oriented design for multicast networks, the Steiner-tree-based design has a moderate improvement in terms of the topology cost, but the network-coding-based design can make this improvement very significant. For example, for the 16-node topology design problem examined in this paper, the Steiner-tree-based design can reduce the topology cost by about 8.3 percent than the conventional unicast-oriented design when the multicast traffic accounts for 50 percent of the total traffic, but our network-coding-based design can make this reduction in topology cost as high as 16.6 percent.

Finally, although two algorithms were proposed for this new topology design problem, how to further reduce their time complexity for the efficient design of large-scale multicast networks is an interesting future work. In addition, the quantitative analysis on the optimal solution and the performance of topology design algorithms deserve further investigation and can also be a subject of future research.

ACKNOWLEDGMENTS

The authors would like to thank the associate editor and the anonymous reviewers for their many valuable comments that helped to considerably improve the paper. This work was supported in part by JSPS Grant-In-Aid of Scientific Research (C) 19500050 and by GCOE Projects at Tohoku University. Dr. Guo's work was partially supported by the China National Natural Science Foundation (No.60533040), the National High Technology Research and Development Program of China (No. 2006AA01Z199), and the Shanghai Pujiang Plan (No. 07pj14049).

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, "Network Information Flow," *IEEE Trans. Information Theory,* vol. 46, no. 4, pp. 1204-1216, July 2000.

[2] Z. Li and B. Li, "Network Coding: The Case of Multiple Unicast Sessions," *Proc. 42nd Ann. Allerton Conf. Comm., Control, and Computing,* Sept. 2004.

[3] D.S. Lun, M. Médard, and M. Effros, "On Coding for Reliable Communication over Packet Networks," *Proc. 42nd Ann. Allerton Conf. Comm., Control, and Computing,* Sept. 2004.

[4] D.S. Lun, M. Médard, and R. Koetter, "Network Coding for Efficient Wireless Unicast," *Proc. Int'l Zurich Seminar Comm. (IZS '06),* Feb. 2006.

[5] D.S. Lun, N. Ratnakar, M. Médard, R. Koetter, D.R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-Cost Multicast over Coded Packet Networks," *IEEE Trans. Information Theory,* vol. 52, no. 6, pp. 2608-2623, June 2006.

[6] C. Chekuri, C. Fragouli, and E. Soljanin, "On Average Throughput and Alphabet Size in Network Coding," *IEEE Trans. Information Theory,* vol. 52, no. 6, pp. 2410-2424, June 2006.

[7] M. Charikar and A. Agarwal, "On the Advantage of Network Coding for Improving Network Throughput," *Proc. IEEE Information Theory Workshop (ITW '04),* Oct. 2004.

[8] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale Content Distribution," *Proc. IEEE INFOCOM '05,* pp. 2235-2245, May 2005.

[9] C. Fragouli, J. Boudec, and J. Widmer, "Network Coding: An Instant Primer," *ACM SIGCOMM Computer Comm. Rev.,* vol. 36, no. 1, pp. 63-68, Jan. 2006.

[10] A. Kershenbaum, *Telecommunications Network Design Algorithms.* McGraw-Hill, 1993.

[11] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[12] K. Steiglitz, P. Weiner, and D.J. Kleitman, "The Design of Minimum Cost Survivable Networks," *IEEE Trans. Circuit Theory,* vol. 16, no. 4, pp. 455-460, Nov. 1969.

[13] M. Gerla and L. Kleinrock, "On the Topological Design of Distributed Computer Networks," *IEEE Trans. Comm.,* vol. 25, no. 1, pp. 48-60, Jan. 1977.

[14] A. Kershenbaum, P. Kermani, and G.A. Grover, "MENTOR: An Algorithm for Mesh Network Topological Optimization and Routing," *IEEE Trans. Comm.,* vol. 39, no. 4, pp. 503-513, Apr. 1991.

[15] F. Glover, M. Lee, and J. Ryan, "Least-Cost Network Topology Design for a New Service: An Application of a Tabu Search," *Annals of Operations Research,* vol. 33, pp. 351-362, 1991.

[16] S. Pierre, M.-A. Hyppolite, J.-M. Bourjolly, and O. Dioume, "Topological Design of Computer Communication Networks Using Simulated Annealing," *Eng. Applications of Artificial Intelligence,* vol. 8, no. 1, pp. 61-69, 1995.

[17] S. Pierre and G. Legault, "An Evolutionary Approach for Configuring Economical Packet Switched Computer Networks," *Artificial Intelligence in Eng.,* vol. 10, no. 2, pp. 127-134, 1996.

[18] M. Grötschel, C.L. Monma, and M. Stoer, "Design of Survivable Networks," *Handbook in Operations Research and Management Science,* vol. 7, pp. 617-671, 1995.

[19] D.S. Lun, M. Médard, T. Ho, and R. Koetter, "Network Coding with a Cost Criterion," *Proc. Int'l Symp. Information Theory and Its Applications (ISITA '04),* Oct. 2004.

[20] D. Bertsekas and R. Gallager, *Data Networks,* second ed. Prentice Hall, 1992.

[21] K. Bhattad, N. Ratnakar, R. Koetter, and K.R. Narayanan, "Minimal Network Coding for Multicast," *Proc. IEEE Int'l Symp. Information Theory (ISIT '05),* pp. 1730-1734, Sept. 2005.

[22] Y. Wu, P.A. Chou, and S.-Y. Kung, "Minimum-Energy Multicast in Mobile Ad Hoc Networks Using Network Coding," *IEEE Trans. Comm.,* vol. 53, no. 11, pp. 1906-1918, Nov. 2005.

[23] R. Bolla, R. Bruschi, F. Davoli, and M. Repetto, "Analytical/Simulation Optimization System for Access Control and Bandwidth Allocation in IP Networks with QoS," *Proc. Int'l Symp. Performance Evaluation of Computer and Telecomm. Systems (SPECTS '05),* pp. 339-348, July 2005.

[24] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and Analysis of Single-Hop Delay on an IP Backbone Network," *IEEE J. Selected Areas in Comm.,* vol. 21, no. 6, pp. 908-921, Aug. 2003.

[25] K. Ishibashi, R. Kawahara, T. Asaka, M. Aida, S. Ono, and S. Asano, "Detection of TCP Performance Degradation Using Link Utilization Statistics," *IEICE Trans. Comm.,* vol. E89-B, no. 1, pp. 47-56, Jan. 2006.
[26] B. Jackson and T. Jordan, "A Near Optimal Algorithm for Vertex Connectivity Augmentation," *LNCS 1969,* pp. 312-325, 2000.
[27] S.-C. Fang and S. Puthenpura, *Linear Optimization and Extensions: Theory and Algorithms.* Prentice Hall, 1993.
[28] D.W. Corne, M.J. Oates, and G.D. Smith, *Telecommunications Optimization: Heuristic and Adaptive Techniques.* John Wiley & Sons, 2000.
[29] M. Listanti, V. Eramo, and R. Sabella, "Architectural and Technological Issues for Future Optical Internet Networks," *IEEE Comm. Magazine,* vol. 38, no. 9, pp. 82-92, Sept. 2000.
[30] M. Charikar, C. Chekuri, T.-Y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation Algorithms for Directed Steiner Problems," *J. Algorithms,* vol. 33, no. 1, pp. 73-91, 1999.

Kaikai Chi received the BS and MS degrees from Xidian University, Shaanxi, China, in 2002 and 2005, respectively. He is now a PhD candidate in the Graduate School of Information Sciences, Tohoku University, Sendai, Japan. His current research focuses on the application of network coding in wired networks and wireless networks, such as the topology design of network-coding-based networks and the design of routing algorithms in network-coding-capable networks. He is a student member of the IEEE.

Xiaohong Jiang received the BS, MS, and PhD degrees from Xidian University, Xi'an, China, in 1989, 1992, and 1999, respectively. He is currently an associate professor in the Department of Computer Science, Graduate School of Information Science, Tohoku University, Japan. Before joining Tohoku University, he was an assistant professor in the Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), from October 2001 to January 2005. He was a Japan Society for the Promotion of Science (JSPS) postdoctoral research fellow at JAIST from October 1999 to October 2001. He was a research associate in the Department of Electronics and Electrical Engineering, University of Edinburgh, from March 1999 to October 1999. His research interests include optical switching networks, routers, network coding, WDM networks, VoIP, interconnection networks, IC yield modeling, timing analysis of digital circuits, clock distribution, and fault-tolerant technologies for VLSI/WSI. He has published more than 100 referred technical papers in these areas. He is a member of the IEEE.

Susumu Horiguchi received the BEng, MEng, and PhD degrees from Tohoku University in 1976, 1978, and 1981, respectively. He is currently a professor and the chair of the Department of Computer Science, Graduate School of Information Science, and the chair of the Department of Information Engineering, Faculty of Engineering, Tohoku University. He was a visiting scientist at the IBM T.J. Watson Research Center from 1986 to 1987. He was also a professor in the Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). He has been involved in organizing international workshops, symposia, and conferences sponsored by the IEEE, IEICE, IASTED, and IPS. He has published more than 150 technical papers on optical networks, interconnection networks, parallel algorithms, high-performance computer architectures, and VLSI/WSI architectures. He is a senior member of the IEEE and member of IPS and IASTED.

Minyi Guo received the PhD degree in computer science from the University of Tsukuba, Japan. Before 2000, he had been a research scientist of NEC Corp., Japan. He is now a chair professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He was also a visiting professor at Georgia State University, Hong Kong Polytechnic University, the University of Hong Kong, the National Sun Yet-sen University, Taiwan, the University of Waterloo, Canada, and the University of New South Wales, Australia. He is also a professor at the University of Aizu, Japan. He has published more than 120 research papers in international journals and conference proceedings. He has served as the general chair and the program committee or organizing committee chair for many international conferences. He is the founder of the International Conference on Parallel and Distributed Processing and Applications (ISPA) and the International Conference on Embedded and Ubiquitous Computing (EUC). He is the editor-in-chief of the *Journal of Embedded Systems*. He is also on the editorial board of the *Journal of Pervasive Computing and Communications*, the *International Journal of High Performance Computing and Networking*, the *Journal of Embedded Computing*, the *Journal of Parallel and Distributed Scientific and Engineering Computing*, and the *International Journal of Computer and Applications*. His research interests include parallel and distributed processing, parallelizing compilers, pervasive computing, embedded software optimization, molecular computing, and software engineering. He is a senior member of the IEEE and a member of the ACM, IPSJ, and IEICE.